

Machine Learning

Linear Regression and Regularization

DSC 240

Feb 1, 2025

Instructor: Prof. Yu-Xiang Wang

Announcement

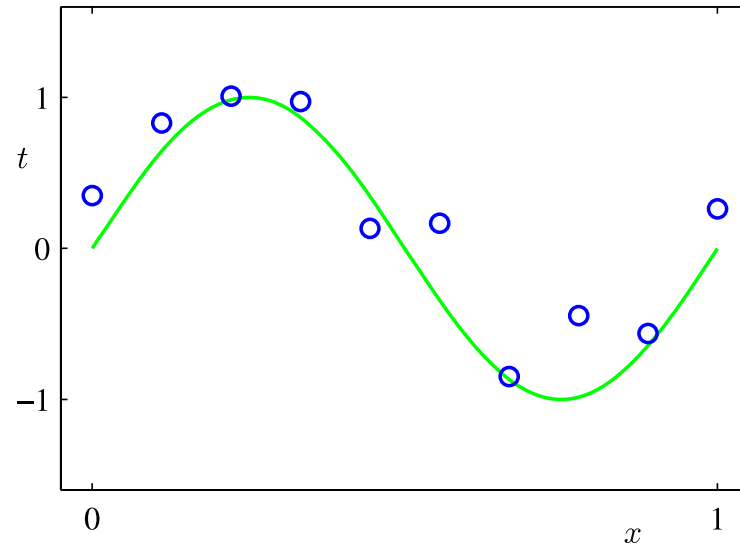
- MP2 out (due in 2 weeks)
- Homework 2 out (due in 2 weeks)
- Midterm next Thursday in class
 - 75 min
 - One “cheat sheet” is allowed: two-sided letter size
 - Topic: classification, decision boundary, loss minimization, and continuous optimization algorithms

Last lecture

- Stochastic Gradient Descent
 - Perceptron (in the offline case when we sample data points at random) is SGD with shifted hinge loss with learning rate = 1.
 - Convergence theorems and how to choose learning rate.
 - “Learning curves” --- how do you debug SGD (or other iterative learning / optimization algorithms)
- Regression problems
 - What is the output space
 - What are good performance metrics
 - Prediction vs Estimation

Recap: Example 1 Curve fitting

Figure 1.2 Plot of a training data set of $N = 10$ points, shown as blue circles, each comprising an observation of the input variable x along with the corresponding target variable t . The green curve shows the function $\sin(2\pi x)$ used to generate the data. Our goal is to predict the value of t for some new value of x , without knowledge of the green curve.



- Input: $x \in \mathcal{X} = [0, 1] \subset \mathbb{R}$
- Output: $y \in \mathcal{Y} = \mathbb{R}$
- Hypothesis: $f(x) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M$
- Data: $(x_1, y_1), \dots, (x_n, y_n)$
- Ground truth: $f_0(x) = \sin(2\pi x)$

Recap: The goal of a regression problem could be (A) Prediction (B) Estimation

- **Prediction task** aims at predicting y using x .
 - Mean square loss (also called “mean square error of a predictor \hat{f} ”)

$$\frac{1}{n} \sum_{i=1}^n (\hat{f}(x_i) - y_i)^2$$

Quiz: Should the above be on training data or test data?

- **Estimation task** aims at making inference about the (unobserved) f_0
 - Mean square error of an estimator \hat{f}

$$\frac{1}{n} \sum_{i=1}^n (\hat{f}(x_i) - f_0(x_i))^2$$

- Requires assumptions on how y_i is related to $f_0(x_i)$

Recap: How to solve “curve fitting” given a hypothesis class?

- The distinction (esp. for square loss) does not matter very much. Challenge:
 - We don’t have access to future data for prediction!
 - We also don’t have access to ground truth f_0
- By solving an optimization problem that **minimizes the loss function on the training data**, and hope that it generalizes.
 - Notice that we can verify if it generalizes or not using hold-out / cross-validation...
- The “least square” objective function:

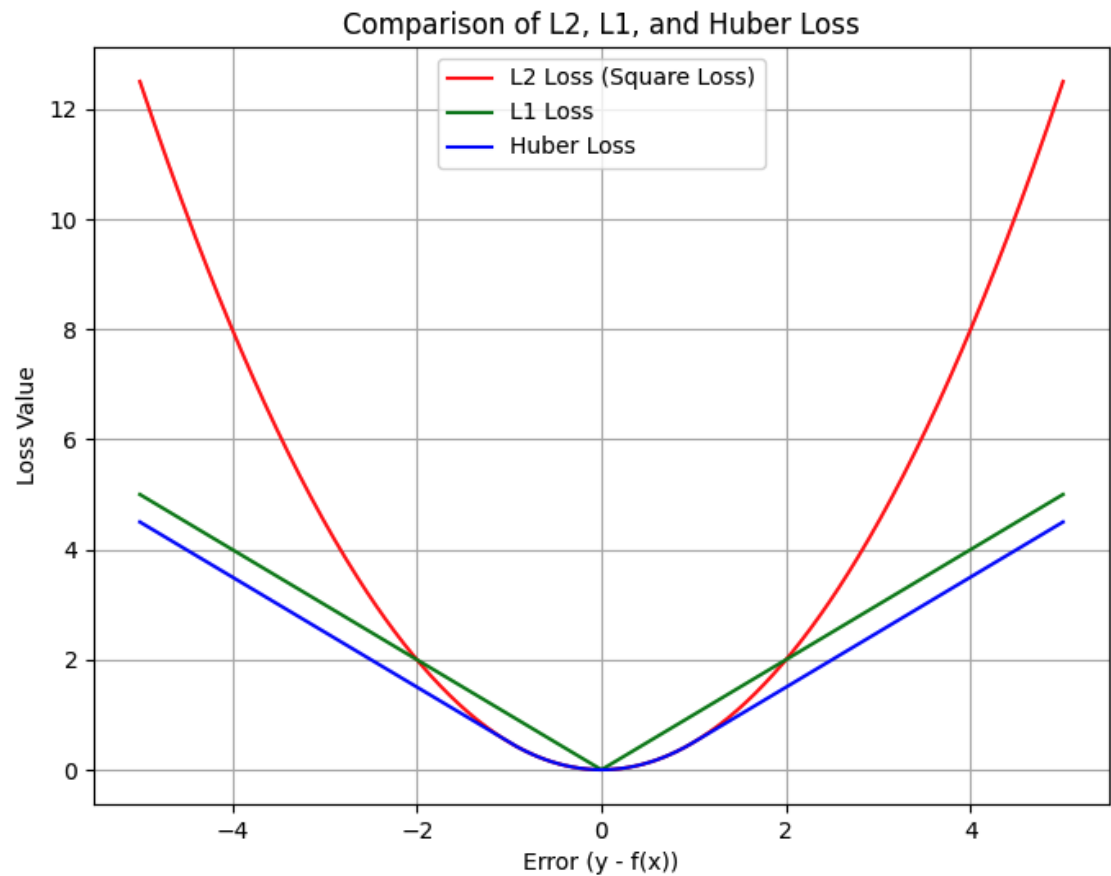
$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2$$

Recap: Examples of supervised learning problems

	Binary classification	Multi-class classification	Regression
Feature space	\mathbb{R}^d	\mathbb{R}^d	\mathbb{R}^d
Label space	$\{-1, 1\}$	$\{1, 2, 3, \dots, K\}$	\mathbb{R}
Popular Performance metric	Classification error (0-1 loss) for test data	Classification error (0-1 loss) for test data	Mean Square Error (against ground truth or against labels)
Popular surrogate loss (for training)	Logistic loss	Multiclass logistic loss aka. Cross-Entropy loss	Square loss

Different losses for regression problems

- Square loss
- L1-loss (Mean Absolute Deviation loss)
- Huber-loss
- And many more...



Which one to use? Depend on applications and computational tractability.

Today

- Linear regression
 - Solving the Least Square problem
 - Regularization
- Case study: Predict California Housing Market

“Regression” example 2: Linear regression

- Case: California Housing Dataset

- 8 features:
 - MedInc median income in block group
 - HouseAge median house age in block group
 - AveRooms average number of rooms per household
 - AveBedrms average number of bedrooms per household
 - Population block group population
 - AveOccup average number of household members
 - Latitude block group latitude
 - Longitude block group longitude
- 1 label: Median house value in \$100,000.

- Feature space
- Label space
- Hypothesis space
- Loss function

Polynomial “Curve fitting” can be casted as a linear regression problem.

- What are the features?

- What is the linear score function?

The objective function for learning linear regression under square loss

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i \in [n]} (x_i^T \theta - y_i)^2$$

- aka: Ordinary Least square (OLS)
- There is a convenient form using linear algebra

Exercise: What's the GD update rule?

- GD update rule?

- Apply linear algebra to simplify the expression in matrix form.

Direct solution to the linear regression problem by solving a linear system of equations

- Recall the matrix form of the gradient?

- How to solve an optimization problem by hand?
 - In the univariate case: Set derivative to 0
 - In the multi-variate case: we can set gradient to 0.
 - This returns the optimal solution if the problem is convex.

Comparing SGD and the direct solver

- Time complexity of direct solver
 - The smaller of $O(d^2 n)$ or $O(d n^2)$.
 - This is $O(n^3)$ when d and n are on the same order.
- Time complexity
 - GD: $O(\min(dn, d^2) * \text{number of iterations})$
 - SGD $O(d * \text{number of iterations})$

Checkpoint: linear regression basics

- The prediction task with performance measured in square loss.
- The least square objective function
 - Can be solved using GD, SGD or directly solving a linear system of equations.
- Up next: regularization

Recap: Polynomial regression under square loss

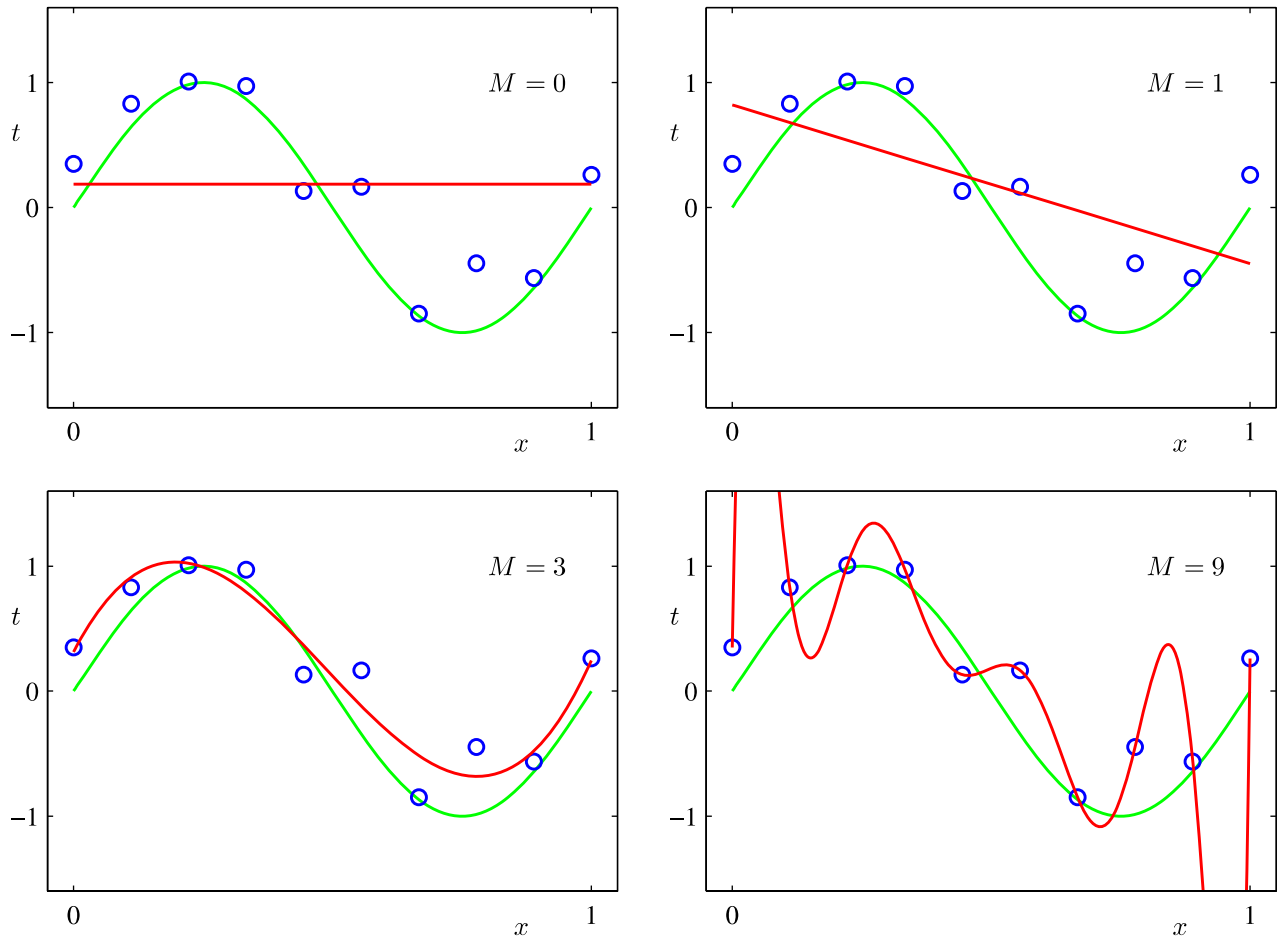
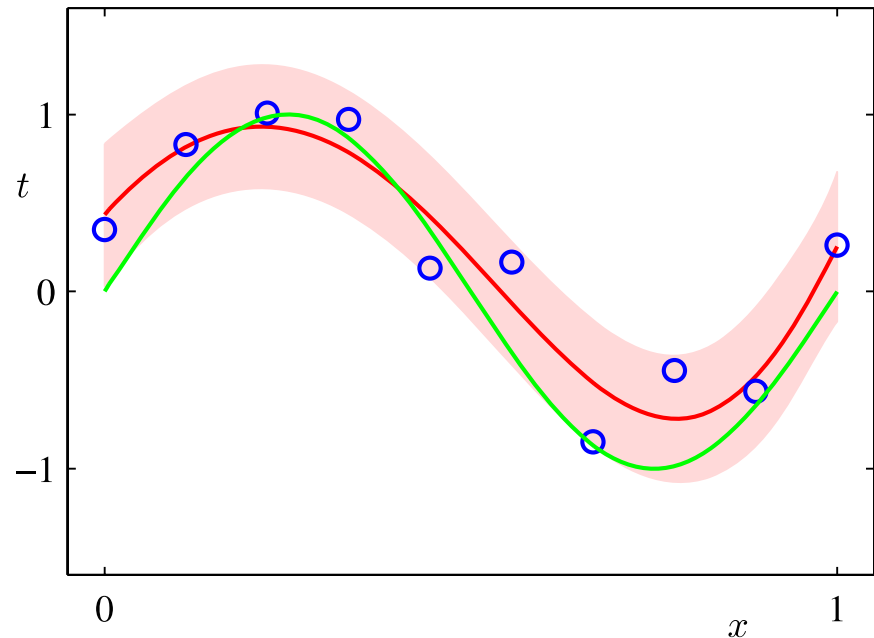


Figure 1.4 Plots of polynomials having various orders M , shown as red curves, fitted to the data set shown in Figure 1.2.

Appropriately regularized fit of a 9th order polynomial.

Figure 1.17 The predictive distribution resulting from a Bayesian treatment of polynomial curve fitting using an $M = 9$ polynomial, with the fixed parameters $\alpha = 5 \times 10^{-3}$ and $\beta = 11.1$ (corresponding to the known noise variance), in which the red curve denotes the mean of the predictive distribution and the red region corresponds to ± 1 standard deviation around the mean.



Regularization prevents overfitting!

Regularization helps to **reduce overfitting** and induce **structures** in the solution.

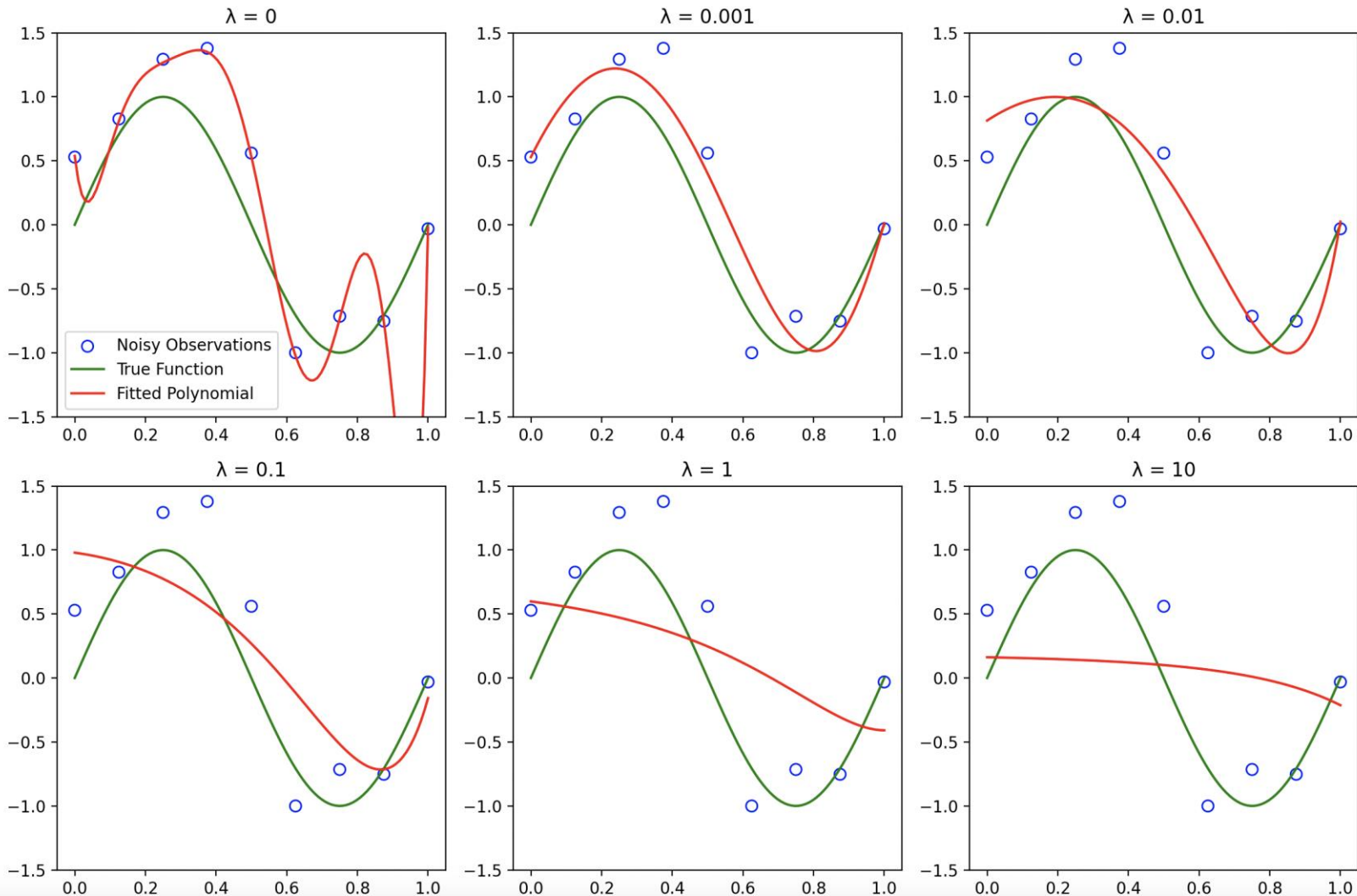
- Example: p-norm regularized least square

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \|X\theta - y\|_2^2 + \lambda \|\theta\|_p^p$$

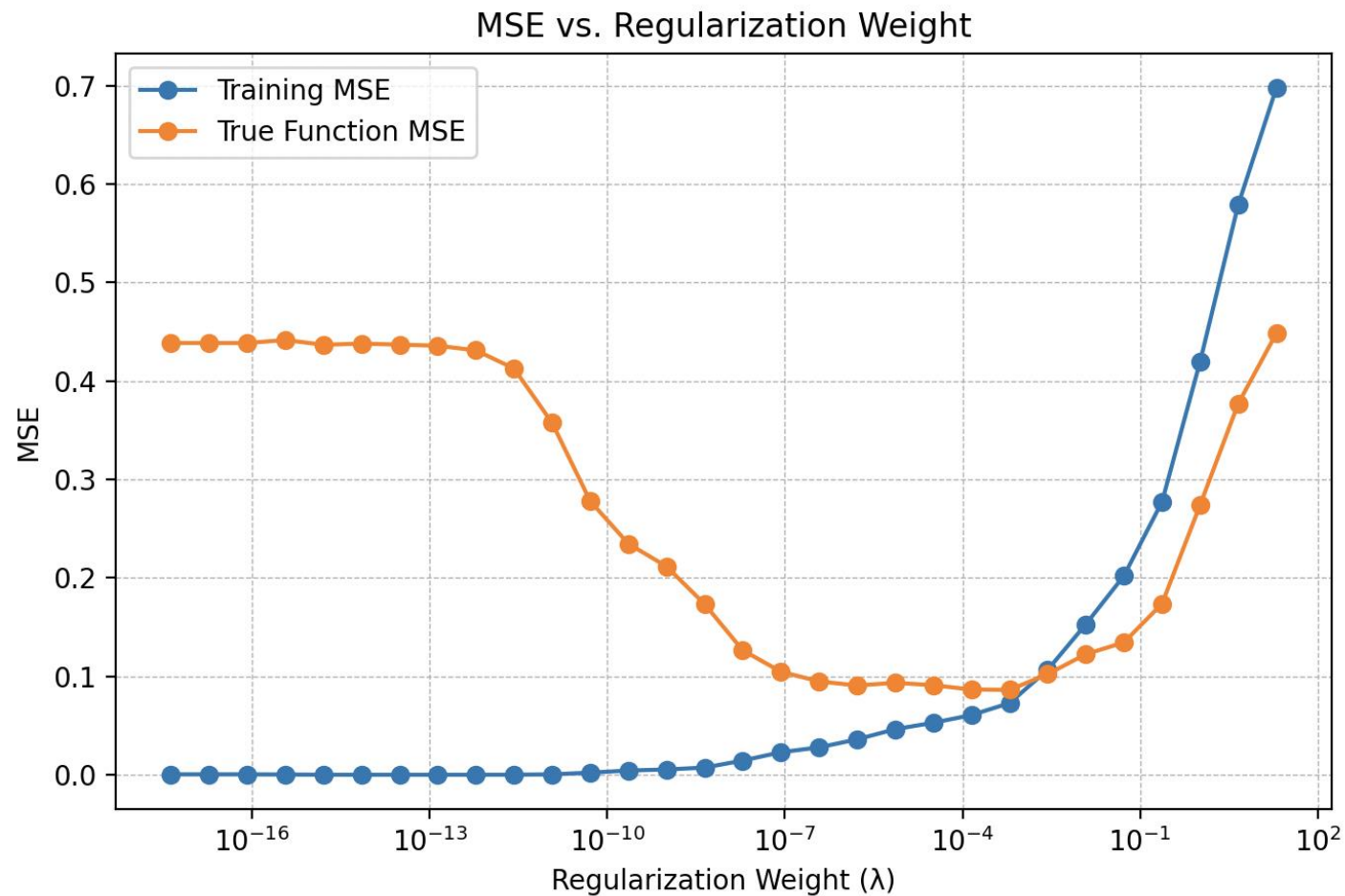
- when p=2, this is called “Ridge Regression”
- when p=1, this is called “Lasso”
- when p=0, this is called “Best subset (feature) selection”

(you will work out closed-form solution for p=2 in the homework)

Fitted polynomial as L2 regularization weight increases



The mean square errors as we adjust the L2 regularization weight



Why a U-shape curve? Bias and variance tradeoff.

$$\mathbb{E}[(f(x) - f_0(x))^2] = \mathbb{E}[(f(x) - \mathbb{E}f(x))^2] + (\mathbb{E}f(x) - f_0(x))^2$$

- If we assume zero-mean independent noise data assumption

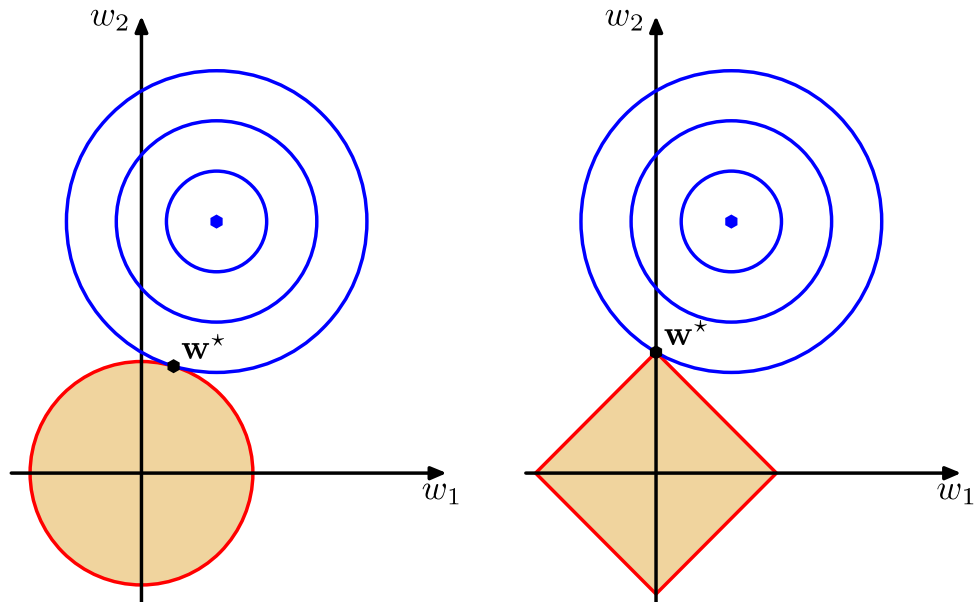
$$\mathbb{E}[(f(x) - y)^2] = \mathbb{E}[(f(x) - \mathbb{E}f(x))^2] + (\mathbb{E}f(x) - f_0(x))^2 + \mathbb{E}[(f_0(x) - y)^2]$$

- Increase regularization weight => Increase bias, reduce variance
- Decrease regularization weight => Decrease bias, increase variance.

Regularization helps to **reduce overfitting** and induce **structures** in the solution.

- Ridge regression induces solutions that are small but dense.
- Lasso induces solutions that are “sparse”.

Figure 3.4 Plot of the contours of the unregularized error function (blue) along with the constraint region (3.30) for the quadratic regularizer $q = 2$ on the left and the lasso regularizer $q = 1$ on the right, in which the optimum value for the parameter vector w is denoted by w^* . The lasso gives a sparse solution in which $w_1^* = 0$.



Case study: California Housing dataset

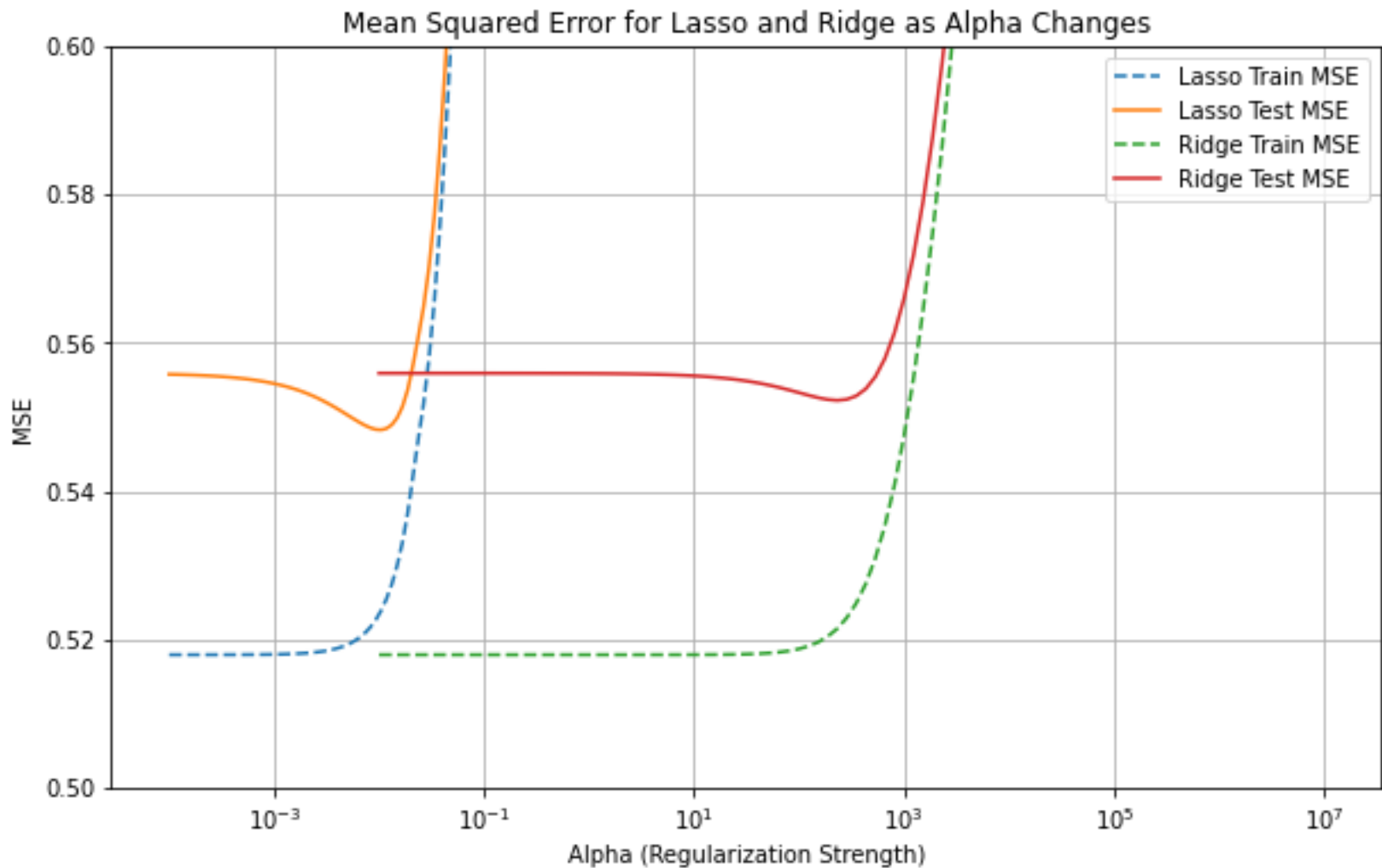
- Example data

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	Target
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422

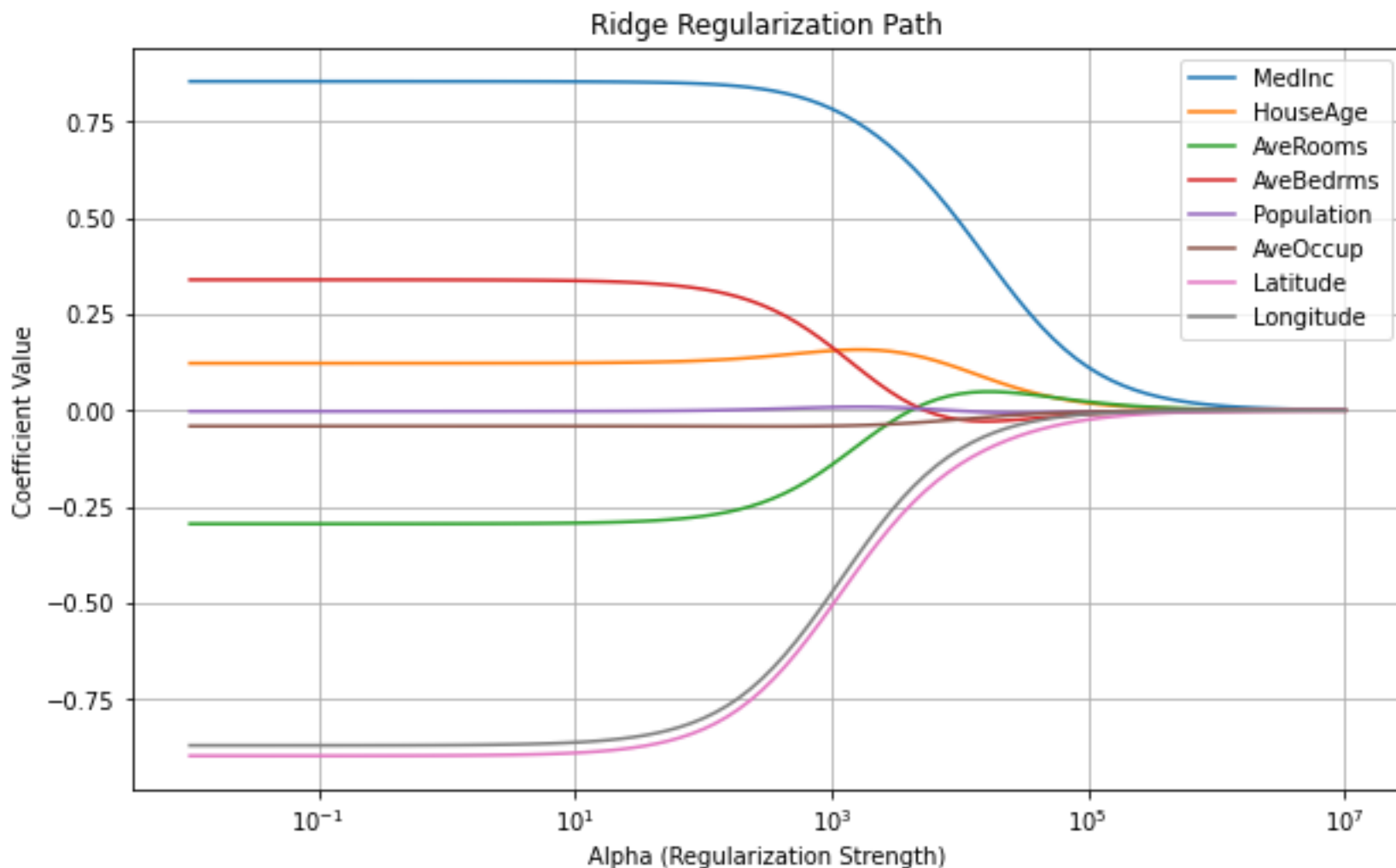
- Questions one can ask:

- How well can I use the 8 features to predict sales price (i.e., Target)?
- Which feature is more predictive with sales price?
- What is the effect of regularization?

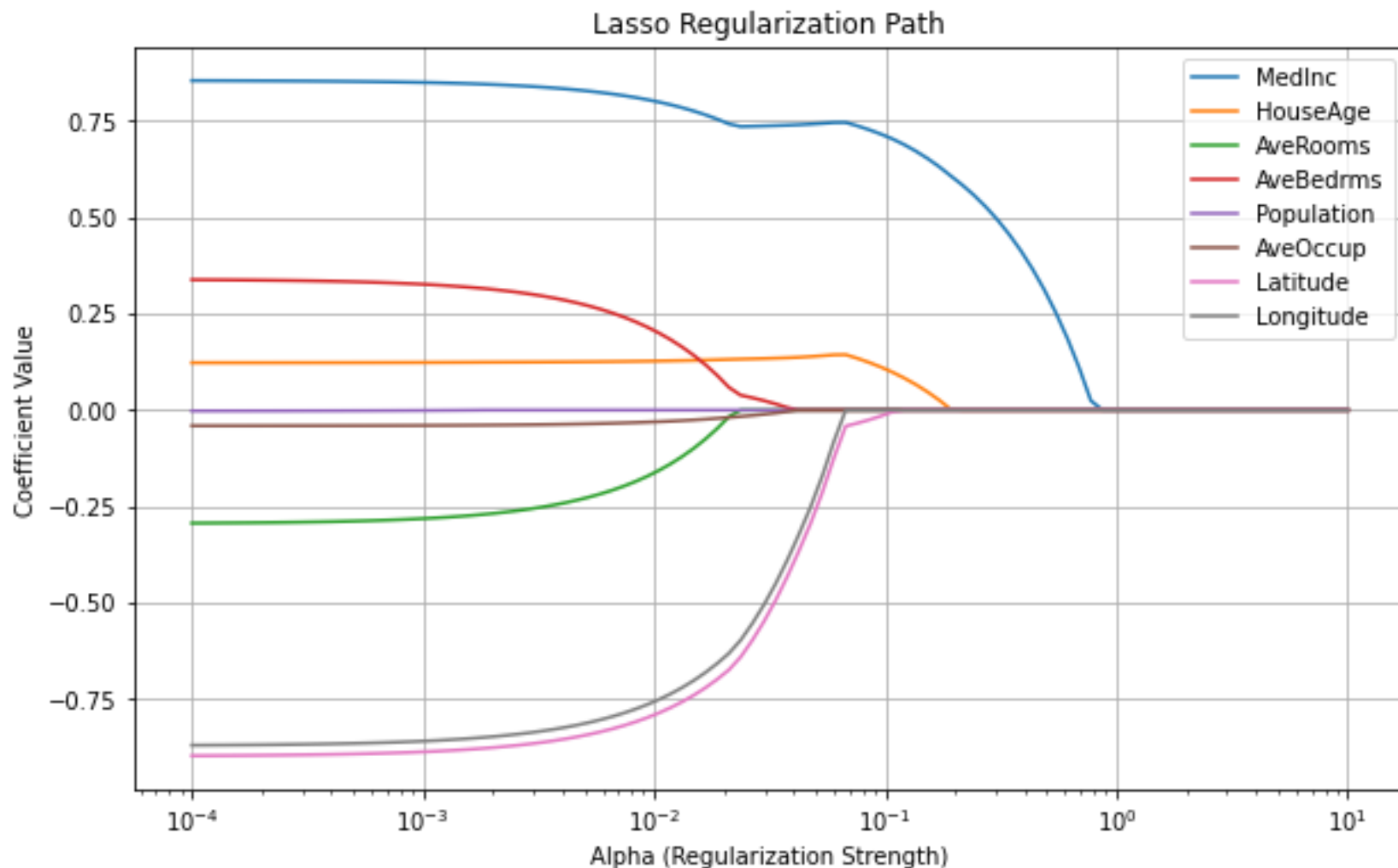
The MSE vs regularization weight plot



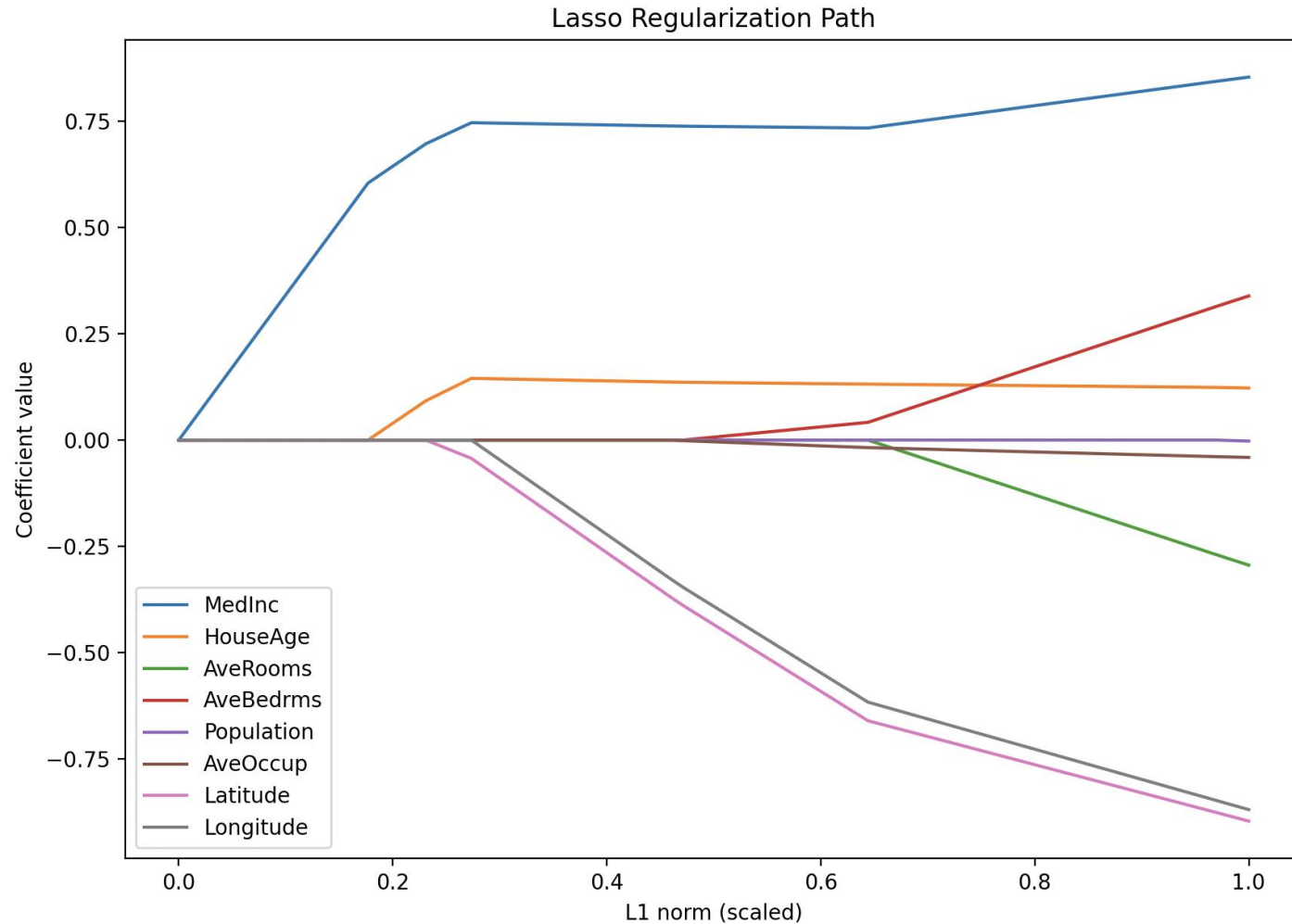
The “Regularization path” for L2-regularization



The “Regularization path” for L1-regularization



Regularization path for Lasso (changing the x-axis to the L1-norm of coefficients)



How to interpret the fitted coefficients?

- The “sign” indicates positive or negative correlation with the label
- The “magnitude” indicates how strongly correlated.
- More precise quantification can be done via statistical inference
 - Hypothesis testing
 - Confidence intervals

(Warning: More assumptions needed for these inference.)

Summary of today's lecture

- Linear regression
 - Solving the Least Square problem {with GD, SGD and direct solver}
 - Regularization {with L2 and L1 regularization}
 - On curve fitting example: L2 regularization controls the regularity of the fitted function --- bias-variance tradeoff.
- Case study: Predict California Housing Market
 - Bias-variance tradeoff on a real dataset
 - Effect of regularization and Regularization path

Next lecture

- Midterm review
- Probability and Statistics